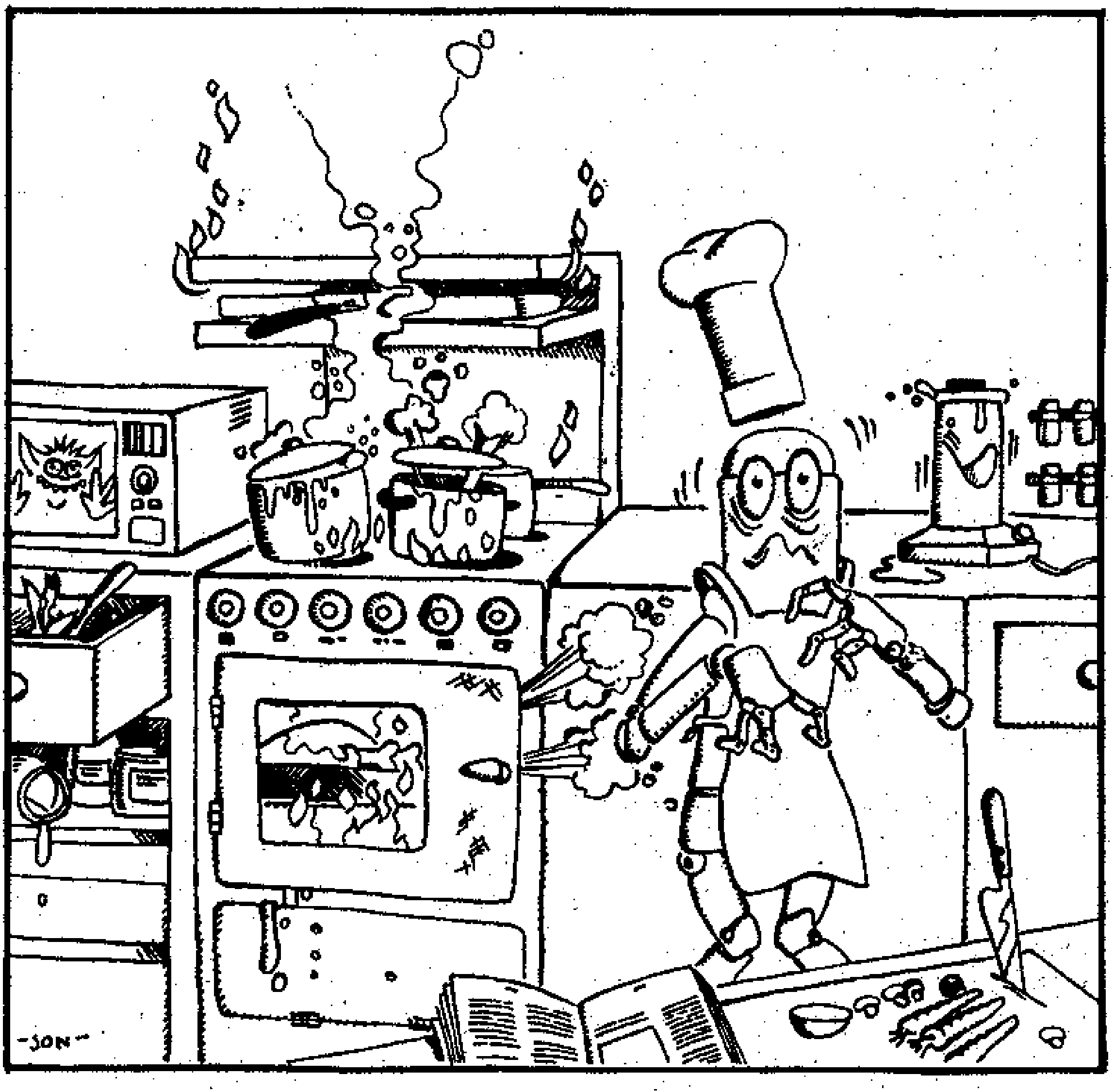
Vol 2 - No 7 February 1989.

FORMAT

THE MONTHLY MAGAZINE FOR SPECTRUM, DISCiPLE & PLUS D USERS



Put Your Spectrum To Work In The Kitchen

Vol 2 No 7

CONTENTS

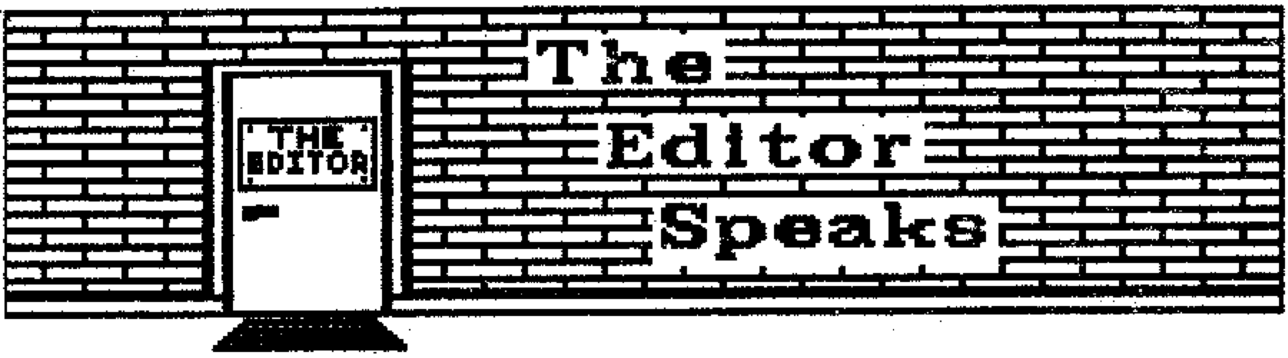
Feb 1989.

THIS MONTHS ADVERTISERS:-

BETTERBYTES	Back Cover
BRADWAY SOFTWARE	8
KEMSOFT	31
P.C.G.	14

FORMAT is published by INDUG. 34 Bourton Road, Gloucester, GL4 OLE, England. Telephone 0452-412572. **DISCiPLE** and **PLUS D** are trade marks of MILES GORDON TECHNOLOGY, Lakeside Technology Park, Phoenix Way, Swansea, South Wales, SA7 9EH. Telephone 0792-791100. The DISCiPLE is marketed by Rockfort Products, 81 Church Road, Hendon, London, NW4 4DP.

Printed by D.S.LITHO. Gloucester. Tel:- (0452) 23198



The Editor Speaks

Space is very short at the moment, yet again several items have had to be left out (and a few others shortened to fit). I did ask last month for a volunteer to help with getting more advertising into FORMAT but my plea seems to have fallen on deaf ears (or does no one read my editorial?). More adverts would equal more pages of articles so come on, someone lend a helping hand. At the same time, if you know of any software company still trading then let me know, FORMAT is bound to be the best place for them to advertise.

The next item this month is to announce a new regular feature for FORMAT. Called 'SHORT STOP' it is a section devoted to small routines either in basic or machine code. These could be subroutines, functions or small programs complete in themselves. John Wase is going to be compiling the page (or 2) and contributions should be sent to him direct. His address is:- Green Leys Cottage, Bishampton, Pershore, Worcs, WR10 2LX. Remember to include a stamped addressed envelope if you want your work returned. We will have the first page as soon as he gets enough in to fill it.

Its not very often I make mistakes in FORMAT (OK, I cant spell, but I dont count those sort of mistakes) but last month I made a big one. I must say sorry to Dick Guy for totally messing up his RS232 article. What did I do wrong? Well anyone trying to build the interface (and I know from the phone calls and letters there are many of you) found I had left out the components list. I plead overwork, not that that will do me much good, in squeezing last months articles into the space available I lost the list from the end of the article. Still better late than never, you will find the parts list on page 31.

Anyone who has registered an interest in SAM should be receiving a mail out from MGT over the next few weeks. There now seens to be some delay in the expected release date of SAM but I will have full news for you next month. Work loads on Alan Miles has also meant a delay until next month of details of their INNOVATION AWARD.

Finally this month a look into the past. I dont like to harp on about things too much but it does seem that when I ask people to send things in everyone sits there and thinks "Oh someone will send Bob what he wants, so there's no need for me to bother". Now when you all think the same thing I dont get anything. So what about details of Computer Clubs (Vol 1/11); samples of Computer Art (1/8); Hints & Tips (several times). Come on, If you dont write in you cant expect others to.

See you next month. **Bob Brenchley. Editor.**

NEWS ON 4

SAM DEBUT IN BLACKPOOL

The SAM Coupe received its public debut at the NARSA show on Sunday 29th January. MGT had a large stand in the main hall and there was a constant large crowd around the stand from just after the show opened until well after it had officially closed.

The prototype SAM was hidden from view but was used to run a varied selection of demo routines that caused many people to doubt that it was SAM and not an Arcamedis (It was SAM, honest, there were no tricks).

A wooden - designers model - of the case and keyboard was also displayed on the stand. Many visitors were able to give their feelings and comments about the new machine. Alan Miles and Bruce Gordon were very pleased with the way SAM had been received. "Our feelings about the appeal of the machine have been comfirmed" said Bruce.

BLACKPOOL SHOW BREAKS RECORDS.

At 10.30am when the doors of the Norbreck Hotel opened the halls of the NARSA show were almost instantly awash with people eager to see what was on offer. By 12 noon there were still queues outside waiting to get in. Unconfirmed reports puts attendance at around the 6000 mark which for a cold Sunday in January would be unbelievable if I had not seen the crowds for myself

Three main halls, plus several side rooms and hallways were packed with stalls offering a wide range of Radio Amateur and Electronic equipment. Many stands featured computer equipment and software and there were several stands devoted just to computers. In short there was something for almost everyone. So, if you missed this one, dont miss next years (March 1990).

MGT LOOKING FOR SOFTWARE WRITERS

MGT are asking independent software writers to get in contact with them about SAM. A manual for software developers will be available in the near future and Alan Miles is keen to see small software companies get in on the ground floor. A meeting to outline MGT's software plans will be arranged when a measure of the interest is obtained.

If you have a proven track record in Z80 software please write to MGT at their Swansea offices (mark the envelope (SAM SOFTWARE WRITERS). Please include details of what you have done in the past and the type of software you are interested in developing for SAM. Alan miles has requested that you write in even if you have already contacted MGT, so that an up-to-date list can be compiled.

- - - - - -

If you have any news items you want to pass on then send them in. Please mark the envelope NEWS in the top left corner.

BEYOND SIMPLE UDGs

By: Clyde Bish.

You may remember in my first article (FORMAT Vol 2 #1) I suggested saving space by holding the value of variables only in the variables area in the computer's memory (called VARS) and not in the program area (PROG) as well. (For those who missed it, the line **19 LET a=5** takes up some 16 bytes of memory space when really all you want is the machine to know that variable 'a' has a value of 5, so type in, with no line number **LET a=5** and press ENTER. You'll get the o.k. message. Now LIST, nothing! Not really surprising, you have no program, but the machine does know the value of 'a'. Prove it by typing **PRINT a** <ENTER> and you'll get 5 on screen).

Back to the idea. I said at the time that you could only use this method for variables which didn't get changed, not for those which are for example strength, score, objects held etc. Actually you can use the same idea for these provided that you reset them before you start again, having just lost your last life by bumping into a Balrog. What you do is write a short program to set up these variables. Something like **- 10 LET s=10: LET l=5: DIM f$(10,10)** where s, l and f$ () are strength, lives and the finds array. Now RUN your program, then delete all the lines, and SAVE it. Yes I know that sounds crazy, but what you'll be saving is just those initial values in VARS. All you have to do now is MERGE this information with the main program when you want to restart, with a line such as **9999 PRINT "You're Dead! rewind tape and play to try again": MERGE "** data" GOTO 10

Three points. Firstly SAVE the data program on the adventure tape at the beginning. It makes it much easier to find when you want to reset. Secondly when you save the main program, make sure the data in VARS is already on board. Thirdly don't ever use **RUN.** You'll lose all the data!

Now let's get back to graphics. The sample program I left you with last time used a set start address (60000) which is OK if you happen to want exactly thirty graphics, but space-wasting if you want less, and tough if you want more. Program 1 is a much more versatile version, allowing you to have as many as you want, and wastes not a byte of space. If you have Program 2 from last time stored then load it in and edit the lines to match. Otherwise you'll have to type it in from scratch.

PROGRAM 1.

```
  15 INPUT "How many pictures?"'i: POKE 23681,i: CLEAR 65535-i*180:
LET a=65535-PEEK 23681*180: LET a1=a
  17 LET g$=CHR$ 22+CHR$ 10+CHR$ 0+"ABCDE"+CHR$ 13+"FGHIJ"+CHR$ 13+"
KLMNO"+CHR$ 13+"PQRST": REM capital letters are udgs
  20 FOR c=1 TO PEEK 23681: INPUT "Title of pic CODE ";(c);"?"'t$
  30 PRINT "LOAD pic CODE ";c: LOAD t$CODE a,160
  40 PRINT "Pic CODE ";(c);" starts at ";a: RANDOMIZE a: POKE 23675,
PEEK 23670: POKE 23676,PEEK 23671: PRINT g$
  50 LET k=0: FOR f=22848 TO 22944 STEP 32
  60 FOR n=0 TO 4: LET k=k+1: POKE f+n,7
  70 INPUT "Atribute value?",v
  80 POKE f+n,v: POKE a+160+k,v
  90 NEXT n: NEXT f: LET a=a+180
 100 PAUSE 10: CLS : NEXT c
 200 INPUT "Title for CODE?", LINE t$
 210 SAVE t$CODE a1,65535-a1
 220 PRINT "Verify": VERIFY t$CODE
```

When you RUN it you'll be asked how many illustrations you want. From this answer the program calculates how far RAMTOP has to be lowered to accomodate these and begins a loop. You load in, in turn each of your stored graphic codes as before, noting the start address of each as it is displayed. You'll need this information later. (You could change the PRINT to LPRINT if you have a printer attached). After each is loaded the picture will appear on screen for you to add the colour attributes character square by square as before. When the last batch of data code has been loaded in and the picture coloured you will be prompted to save the whole combined code block onto tape. This will be ready to load to the correct address when you use it in an adventure.

So much for the preparation. Now for the use. Program 2 is again a more versatile version of the original from last time. You remember that in that original program you had to have your illustration in one place on screen? Not any more. Now it will occupy the next available print line (but don't let it scroll or the colours will be misplaced. The attributes file does not scroll!). The trick lies in line 1000. of the revised program (Program 2). You'll note that it PEEKs (i.e. finds out what is in) address 23689. This is the System Variable (within the machine's "housekeeping" memory) which holds the number of the next available print row on screen, although unfortunately it counts upwards from the bottom. Hence the subtraction sum! It is also used to calculate the position in the Attributes file to start poking in the colour data. You'll note that the print column position is still column 0. This would be difficult (although not impossible) to alter as the CHRS 13 in g$ always moves the print position to the beginning of the next line. You'll also note that the first five characters of the data string f$ are different. These five characters now form the start address of the illustration whose name still follows as before (from character 6). These are the addresses you were told as you ran Program 1.

PROGRAM 2.

```
 100 LET f$="65145A Boot": GOSUB 1000: STOP
1000 LET c=VAL f$( TO 5): RANDOMIZE c: POKE 23675,PEEK 23670: POKE 2
3676,PEEK 23671: LET l=24-PEEK 23689: PRINT AT l,0;g$,f$(6 TO ): LET
 c=c+160: LET a=22528+32*l: FOR f=a TO a+96 STEP 32: FOR n=0 TO 4: P
OKE f+n,PEEK c: LET c=c+1: NEXT n: NEXT f: RETURN
9999 LET g$="ABCDE"+CHR$ 13+"FGHIJ"+CHR$ 13+"KLMNO"+CHR$ 13+"PQRST":
 REM Caps are udgs
```

Also in my last article I left you with a problem. How to make larger illustrations using the UDGs. (User Defined Graphics). Of course the obvious answer is to use more UDGs. This is of course

TABLE A

33	15	91	126	35	34	0	91
111	60	200	38	0	41	41	41
237	75	54	92	9	62	8	50
4	91	58	11	91	50	9	91
58	10	91	50	8	91	62	9
50	5	91	126	35	34	2	91
7	50	6	91	58	5	91	61
32	50	58	4	91	61	32	24
58	14	91	71	58	12	91	79
58	10	91	129	5	32	252	50
10	91	42	0	91	195	238	254
50	4	91	58	13	91	71	58
9	91	128	50	9	91	42	2
91	195	11	255	50	5	91	58
12	91	71	58	9	91	50	7
91	58	13	91	79	197	205	143
255	193	58	7	91	60	50	7
91	13	32	241	58	8	91	60
50	8	91	5	32	221	58	6
91	195	27	255	128	64	32	16
8	4	2	1	58	142	92	238
255	71	58	141	92	160	71	58
8	91	230	248	111	58	7	91
254	192	208	31	31	31	230	31
103	203	28	203	29	203	28	203
29	203	28	203	29	62	88	180
103	58	142	92	166	176	119	58
7	91	71	230	7	246	64	103
120	31	31	31	230	24	180	103
120	23	23	230	224	111	58	8
91	71	31	31	31	230	31	181
111	235	33	135	255	120	230	7
79	6	0	9	70	26	33	6
91	203	70	40	3	176	18	201
47	176	47	18	201			

TABLE B

0	0	0	0	15	8	8	15
0	0	0	7	255	2	2	255
0	0	244	244	244	96	96	240
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
8	15	15	15	15	15	31	31
1	255	255	255	255	255	255	255
48	48	240	48	48	248	152	152
0	0	0	0	0	0	7	59
0	0	0	0	0	224	240	248
31	31	63	63	63	63	63	63
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
251	251	253	254	255	255	255	255
252	252	254	254	2	254	254	254
63	32	63	16	15	0	0	0
255	1	255	1	255	0	0	0
3	248	7	0	0	0	0	0
248	7	240	15	0	0	0	0
2	252	4	248	0	0	0	0

quite possible, but it also uses up more memory. Here's a solution which takes up the same storage space and gives you illustrations twice as large provided you're prepared to accept a slight loss in resolution. (You'll get the sort of quality of illustration found in the "Dangermouse" adventures).

We need to enter the realms of machine code. for the uninitiated this is a form of programming which operates directly on the machine's processor without the need to interpret the instructions line by line. In consequence it is much faster than BASIC. It is also more likely to go wrong as there are no automatic error checks! But fear not. As long as you put in the data as printed all will be well. The data is found in Table A. Input it carefully into your machine, reading across each line, using this short program (which you'll be using often so I'll refer to it in future as "codeloader"):-

```
10 REM Machine in 48k mode
20 CLEAR 59999
30 FOR F=65259 TO 65535:INPUT "Number?",a: POKE F, a: PRINT PEEK F : NEXT F
```

for the moment SAVE the code using **SAVE "BIGCODE" CODE 65259,277** and VERIFY. It would take up too much space to explain exactly how the machine code works. Suffice it to say that it takes each byte of information in turn and enlarges it on screen.

OK. Let's give the routine a try out. First we need an illustration for it to work on. Table B gives the data for a picture of a boot. (And why not a boot. Don't you always find them laying around in adventures?!). Load this data in as before using "codeloader" but altering the FOR and TO values of F to 60000 and 60159. Save it with **SAVE "bootg" CODE 60000,160**

Now to try out the idea. Type in Program 3. Let's have a look at what its going to do before you RUN. Line 10 POKEs into the printer buffer - a good place to store things in 48k mode - values that the "newcode" needs to use. The first two are the required magnification, horizontal and vertical, so the 2 in this case means twice normal size. The numbers 144 to 148 are the codes for the UDGs A to E. The 255 is an end marker. Line 20 sets up the value of s, the data starts less 1152. The reason for the substraction is worth an explanation. The routine looks in the ROM - the computers own memory - for he shape of characters to print. You have to direct it to the UDGs instead, but as the first UDG is code 144 and there are 8 bytes per character the routine has to be directed 144 times 8 = 1152 bytes below the real start. (If you're confused by that take consolation in the fact that it took me ages to work it out!). Line 30 does all the hard work. It resets the character pointer as explained above, sets the row and column pointer each loop, then calls the routine to print that row of UDGs. Each loop prints successively lower rows. Line 40 resets the character pointer.

PROGRAM 3.

```
   1 REM You must CLEAR 65213 & have BIGCODE and bootg CODEs on
         board. RUN 9999 to set Code, then use GOTO 1
   2 LET f$="60000A Boot"
   3 GOSUB 20
   9 STOP
  20 LET s=VAL f$( TO 5)-1152: REM start of graphics data less  1152
  30 FOR f=0 TO 3: RANDOMIZE s+40*f: POKE 23606,PEEK 23670: POKE 236
07,PEEK 23671: POKE 23307,81+f*16: POKE 23306,0: RANDOMIZE USR 65259
: NEXT f
  40 POKE 23606,0: POKE 23607,60
9999 POKE 23308,2: POKE 23309,2: POKE 23310,8: POKE 23311,144: POKE
23312,145: POKE 23313,146: POKE 23314,147: POKE 23315,148: POKE 2331
6,255
```

Now RUN and you'll see a big boot appear centre left.

Well thats all the room I have this month, but I will be back next month to show you how to add colour to your BIG graphics.

STREAMS & CHANNELS

By: Nev Young.

Last month I explained the basics of streams and channels on the Spectrum. This month I want to look at the way the DISCiPLE and PLUS D handle channels to disc files.

The first thing to note is that the stream handling is just the same, as are the entries in the STRMS area, and the same two routines are used for input and output. The difference is in what the channel contains. It starts off the same:-

1 Address of output routine = 8
2 Address of input routine = 8
3 channel type = 'D'

You may notice that the input and output addresses are the same and furthermore they are the address of the main error routine. This means that any input or output on these channels will cause a call to the error routine at location 8. This will cause the DOS to be paged in and the main ROM paged out. What happens next is very cunning. The DOS does not know if this address has been reached by an error or a hook code or an input or output command. So first of all the DOS checks the last value on the machine stack. If you have tried an input or output command the two routines at 5606 (15E6h) and 5618 (15F2h) both call the required input or output routine from an instruction at 5627 (15FBh) so the return address on the stack would be 5630 (15FEh). If this is not the case then the program wanders off to do other things. As it is then all we need to do is call the DOS routine to do the work. But just a moment how do we find this routine? You will remember that DE contains the address of the channel's input or output routine. By adding a 5 to this address we can extract some extra info held in the channel. The next data items in a channel are:-

4 Disc output routine
5 Disc input routine
6 Length of this channel
much more data

So from this we can now call the ROM routine we need to read or write to a disc channel. (OPENTYPE FILES that is).

The format of a disc channel differs for a read and a write file. The complete format of them is given in Fig 1 and Fig 2. If you look carefully you will see that bytes 19 to 29 are copied from the file header bytes 0 to 11 in the disc directory. Also bytes 30 to 38 are from the file directory bytes 211-220 and byte 18 is from directory byte 210. Sadly if you OPEN # a file these values are not set.

You can open ANY type of file on the DISCiPLE or PLUS D and read

```
  0 DEFW Main ROM Output Routine Address = 8
  2 DEFW Main ROM Input Routine Address = 8
  4 DEFB Channel Identifier = 'D'
  5 DEFW DOS Output Routine Address = MCHWR
  7 DEFW DOS Input Routine Address = MCHRD
  9 DEFW Channel Length = 551
 11 DEFB Disc Drive Number
 12 DEFB MFLAG = 0 (Read)
 13 DEFW RPT. Offset of next byte to read from buffer
 15 DEFW MBUFF Offset of buffer from start of channel
 17 DEFB NOT USED
 18 DEFB MSB of File Size (ie number of 64K blocks)
 19 DEFB Directory Descriptor (10 for opentype)
 20 DEFS Filename 10 bytes
 30 DEFB File Type  (HD00)  0=BASIC,  1=NUMBER ARRAY,
      2=$ ARRAY, 3=BYTES. NOT USED with OPENTYPE files
 31 DEFW MID and LOW bytes of file size (HD0B)
 33 DEFW Memory Start Address (HD0D)
 35 DEFW Program Length (HD0F)
 37 DEFW Autostart Address (HD11)
 39 DEFS 510 BYTES OF DATA
550 DEFB Next Sector Number
551 DEFB Next Track Number
```

Fig 1. DISC READ CHANNEL - Size = 551 Bytes.

```
  0 DEFW Main ROM Output Routine Address = 8
  2 DEFW Main ROM Input Routine Address = 8
  4 DEFB Channel Identifier = 'D'
  5 DEFW DOS Output Routine Address = MCHWR
  7 DEFW DOS Input Routine Address = MCHRD
  9 DEFW Channel Length = 787
 11 DEFB Disc Drive Number
 12 DEFB MFLAG = 1 (Write)
 13 DEFW RPT Offset of next free byte in buffer
 15 DEFW MBUFF Offset of buffer from start of channel
 17 DEFB Next Sector )i.e. where this data will be
 18 DEFB Next Track  ) written on the disc.
 19 DEFB Directory Descriptor
 20 DEFS 10 byte Filename
 30 DEFW CNT = Sectors used (HI byte first!)
 32 DEFB Track Number of first sector
 33 DEFB Sector Number of first sector
 34 DEFS FSAM 195 Bytes containing the Sector Bit Map
229 DEFB File Size - High byte (= number 64K blocks)
230 DEFB File Type  (HD00)  0=BASIC,  1=NUMBER ARRAY,
      2=$ ARRAY, 3=BYTES. NOT USED with OPENTYPE files
231 DEFW MID and LOW Bytes of file size (HD0B)
233 DEFW Memory Start Address (HD0D)
235 DEFW Program Length (HD0F)
237 DEFW Autostart Address (HD11)
275 DEFS 510 BYTES OF DATA
```

Fig 2. DISC WRITE CHANNEL - Size = 787 Bytes.

the data held in it either from basic by OPEN # or from machine code using Command Code 70.

In the case of a write file you will see that the entire disc file header is held in the channel area in the same sequence as it is held on the disc. If you alter the values held in the channel data you can open a file as an OPENTYPE and close it as a BASIC program, or anything else you can think of.

When you read a byte from a file the file size is decremented and on reaching 0 an end of file report is produced. Also as each byte is read the value held in RPT is incremented and on reaching 510 the next sector is read from disc. (Or the system hangs if there is no next sector). The opposite of these happen when writing a file, in addition as each sector is written a bit is set in the Sector Map to mark it as used. When you close a write file the channel info from byte 19 to 238 is copied to the disc directory.

A word of caution there is a bug in the DOS OPEN # routine that happens both from basic and from hook codes. The sequence of events is:-

1, Create space for new channel.
2, Read the directory for the
   file header.
3, Fill the chans area.
4, Set up STRMS.

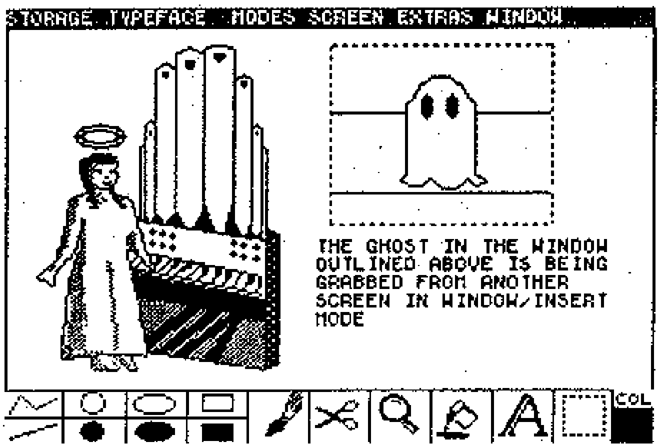
The bug occurs if you press the break key during the directory search. This is detected by the DOS and gives a Break requested message. The space for the new channel exists but the pointers are wrong. So the next attempt to do anything with a DOS stream will hang the machine.

The moral is don't press break when opening a file.

ARTIST II ON DISC

By: Carol Brooksbank.

My computing friends all know that my favourite graphics package is THE ARTIST II, 128K. There isn't a lot to choose between the leading graphics programs when it comes to drawing lines, circles and other assorted shapes, filling them with patterns, and so on. But there are two features on THE ARTIST II which lift it head and shoulders above the competition.



THE GHOST IN THE WINDOW OUTLINED ABOVE IS BEING GRABBED FROM ANOTHER SCREEN IN WINDOW/INSERT MODE

Fig 1.

One is the Insert feature, available in Window and Cut-and-Paste modes. This allows you to define a window in your screen, load a second screen into memory, scroll it around the window until the portion you want is visible, then grab that portion from the second screen into your current artwork.

Put that together with the Plus D's snapshot screen-saving, and every program you own becomes a useful source of clip-art.

The other great feature of THE ARTIST II is the Pagemaker, which holds eight screens in memory at once, letting you scroll around them freely, grab a 'screenful' from any part of the eight-screen page into the main program, modify it and put it back in the page. Text prepared with THE WRITER wordprocessor can be loaded into the page, illustrations and headlines added, to give a desk-top publishing facility. The printout produces an A4 page.

So, when the Plus D came into my life, the first program I had to get running on it was THE ARTIST II. It is possible to load the tape version and take a snapshot, but transferring the whole program to disc allows one or two handy modifications to be made. The transfer is very straightforward. Put the Spectrum into 128K mode.

1) Enter **MERGE ""**. Play the tape. (Stop it when the first program has loaded - we will take stopping it again for granted from now on)
2) Alter the BASIC to give the listing in Program 1.
3) Enter **SAVE D1"ARTIST2" LINE 0**
4) Enter **CLEAR 25910**
5) Enter **LOAD "" CODE.** Play the tape
6) Enter **SAVE D1"ART2A" CODE 32768,6912** (This is the loading screen. If you prefer to leave it out, modify the BASIC in Program 1 before you save it, omitting line 25, and omit step 6)
7) Enter **LOAD "" CODE.** Play the tape
8) Enter **SAVE D1"ART2B" CODE 32768,2048**

9) Enter **LOAD "" CODE**. Play the tape
10) Enter **SAVE D1"ART2C" CODE 34816,1728**
11) Enter **LOAD "" CODE**. Play the tape
12) Enter **SAVE D1"ART2D" CODE 39168,768**
13) Enter **LOAD "" CODE**. Play the tape
14) Enter **SAVE D1 "ART2E" CODE 25911,256**
15) Enter **LOAD "" CODE**. Play the tape
16) Enter **SAVE D1 "ART2F" CODE 26096,39440**
17) Press the reset button. Select 128K mode
18) Enter **MERGE ""**. Play the tape
19) Add a new line to the BASIC:-
    1 PAPER 7: INK 0: CLS: POKE @6,1 (This makes the disc catalogue, displayed whenever the program loads or saves to disc, visible. Left alone, the program has ink and paper black! It also makes the screendump and Pagemaker printing routines work, if your printer supports the ESC "*" bit-image graphics mode)
20) Enter **SAVE D1"ART2G" LINE 98**

The whole program has now been transferred to disc, and entering:-**LOAD D1"ARTIST2"** will load and run it. However, there is another useful modification which can be made.

When you print the A4 page from the Pagemaker, you will discover that it is not centred on the page, and no amount of going into BASIC and POKEing @9,8 or LPRINTing CHR$ 27; "l"; CHR$ 8; will persuade the printer to put a margin at the left. This is because the Pagemaker print routine's first action is to initialize the printer, thus losing any codes you may have sent. Fortunately, it is easily modified.

The print routine is in the code block saved as "ART2B". It is wise to reset the Spectrum, to get rid of any rubbish lying around in the memory, **CLEAR 32767**, and then load the block. Entering the following as a direct command will stop the printer from being reset before the page is printed:-

FOR X=34568 TO 34573: POKE X,0: NEXT X

Now re-save the block to disc, replying 'Y' when asked whether you wish to overwrite the original. **SAVE D1"ART2B" CODE 32768,2048**

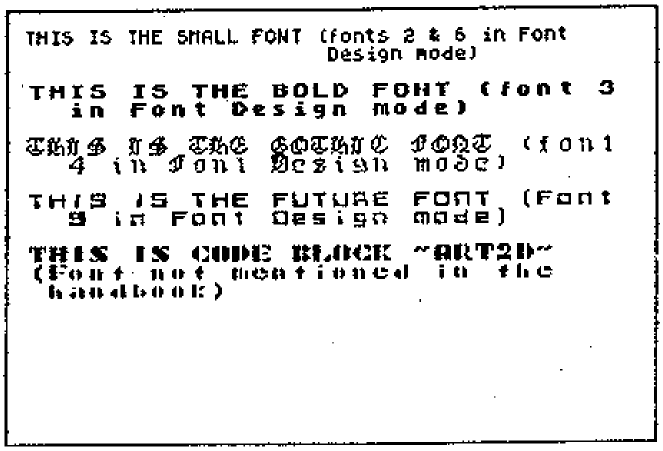
When you run the program, go into BASIC before page printing, and enter the margin setting codes. POKE @9,8 still does not work with this program on my machine, but LPRINT CHR$ 27; "l"; CHR$ 8; does work, and centres the text on the page.

While you still have "ART2B" in memory, you may find it useful to save the Pagemaker printing routine separately:-

SAVE D1"PCOD" CODE 34563,105

If you type in and save to disc the short BASIC program in Program 2, you will have a program, independent of THE ARTIST II, for printing multiple copies of Pagemaker pages. If you use tractor-feed A4 paper, you can go away and do something else while the copies appear. (Make it something long if you want more than a few copies. This is not a speedy program!)

A word or two about the Program 2 BASIC. The inner loop, lines 50 - 100, is needed because the Pagemaker pages are saved in four blocks of code, with 0-3 appended to the page name you chose. The ";" after d1 in line 80 is necessary because without it the Plus D syntax checking will refuse to accept LOAD a$, and will demand the file name written out in full.



THIS IS THE SMALL FONT (fonts 2 & 6 in Font Design mode)

THIS IS THE BOLD FONT (font 3 in Font Design mode)

THIS IS THE GOTHIC FONT (font 4 in Font Design mode)

THIS IS THE FUTURE FONT (Font 5 in Font Design mode)

THIS IS CODE BLOCK ~ART2D~ (Font not mentioned in the handbook)

Fig 2.

Finally, "ART2D" is an extra typeface (see Fig 2) which is not spoken of in the manual. A WRITER file prefaced with the Graphics/7 code will appear in this font when loaded into Pagemaker, but it does not seem to be available to the main program. However, in Font Design mode, "ART2D" can be loaded into either of fonts 3,4 or 5, in place of the original typeface. Font 1 cannot be replaced as it is the Spectrum character set, held in ROM, and you meddle with fonts 2 and 6 at your peril! They hold the small typeface which is availabe for use in your artwork, but which is also used by all the menus. Replace, invert, or mirror it, and the menus turn to gibberish and the cursor becomes invisible. It is virtually impossible either to correct it or to use the program as it stands.

I have not yet been able to get THE WRITER to save and load to disc, or to print, but a snapshot of the tape version can be used to produce the text files for the Pagemaker, saving and loading from tape. Since I cannot get THE WRITER to use the disc, I have not bothered with a bug in THE ARTIST II which refused in the Microdrive version to load WRITER files from the drive. Tape loading works and will do for now.

If ever I get THE WRITER working properly with the Plus D, I will let you know. Or, if you have it up and running, perhaps you will tell me!

PROGRAM 1

```
   1 IF PEEK 75=191 THEN PRINT AT 5,2;"The Artist II will not run on
";AT 7,2;"a 48k Spectrum...": BEEP 2,30: PAUSE 0: NEW
   2 IF PEEK 23388<>16 THEN PRINT AT 5,2;"Please reset and use 128k
mode": BEEP 2,0: PAUSE 0: NEW
  10 PAPER 0: BORDER 0: INK 0
  20 CLEAR 25910
  25 PAPER 8: INK 8: BRIGHT 8: LOAD d1"ART2A"SCREEN$
  26 PRINT AT 0,0;"                                "
  30 POKE 23689,25: LOAD d1"ART2B"CODE
  35 POKE 23689,25: LOAD d1"ART2C"CODE
  40 POKE 23689,25: LOAD d1"ART2D"CODE
  50 POKE 23689,25: LOAD d1"ART2E"CODE
  60 LET c=USR 32771
  70 LET c=USR 25920
  90 POKE 23689,25: LOAD d1"ART2F"CODE
  91 POKE 23606,120: POKE 23607,104
 100 POKE 23689,25: LOAD d1"ART2G"
```

PROGRAM 2

```
 10 CLEAR 34562
 15 POKE @6,1: LPRINT CHR$ 27;"1";CHR$ 8;CHR$ 27;"C";CHR$ 70;
 20 LOAD d1"PCOD"CODE
 30 INPUT "Name of file";n$
 40 INPUT "number of copies";c
 50 FOR x=1 TO c
 60 FOR y=0 TO 3
 70 LET a$=n$+STR$ y
 80 LOAD d1; a$CODE 49152
 90 RANDOMIZE USR 34563
100 NEXT y
110 LPRINT CHR$ 12;
120 NEXT x
130 LPRINT CHR$ 12;
```

* + * + * + * + *

PAGE 701

By: Andy Dunn.

With reference to Volume 2, No.1, page 30. In the second paragraph of P.Anderson's article on Teletext, I notice his item on BBC2 CEEFAX page 701 - the Micro Page. Readers of Format may wish to know a little more about how page 701 "NEXT" - the micro page is compiled.

Martin Cooper is the chief sub-editor at the BBC CEEFAX unit and is in charge of the whole BBC teletext transmissions. NEXT was a micro page that dealt with home micro's like the Amstrad, Atari, Commodore, MSX, BBC and of course the Spectrum. Articles were written by either Martin Cooper or by people sending material to be transmitted.

The feedback derived from NEXT grew out of proportion till finally the bubble burst. In the end it was found that the computer section was taking too much of Martin's valuable time, by answering queries and revamping letters, instead of doing the work which he is paid for, and that is preparing and updating news etc. throughout the BBC1 and BBC2 network.

Pages 700 to 703 were used to give useful hints and tips on computer related software and hardware plus a forthcoming events and club contact spot.

Computer games were also reviewed and Martin Cooper nominated various candidates who had a natural interest in CEEFAX plus being able to write an informative article on a particular review without Martin having to re-spell, edit, or retype an article. In otherwords, the review must be concise and give the viewer a good idea of the software product, its good and bad points, and is its value for money, all in about two teletext pages.

Easy you may think, but a page of teletext is 35 characters long and has 18 usable lines which must include a header and a footer. Therefore the article must be short and to the point.

Keith Hazelton, myself and a few others were nominated to act as reviewers. A wide coverage of different users had to be available depending on what computer they used at home.

Keith and I had the added advantage that although we were Spectrum users, we were also subscribers to PRESTEL. This was a great advantage as we were able to compose and transmit our reviews direct to Martin Cooper via the Prestel mailbox. As the Prestel page format was different to a teletext page - one page of CEEFAX was sent as two PRESTEL mailboxes - the first being lines 1 to 9 and the second mailbox lines 10 to 18.

Martin Cooper at CEEFAX was able to save the text onto his BBC micro and strip off the line numbers, headers and footers and able to splice the two pages together to form one complete page suitable to be inputted into the CEEFAX computer ready for transmission on the next up-date, usually early Saturday morning.

As Keith and I have a very good relationship over Prestel, although we have never met face-to-face, I took the opportunity for INDUG to have their name included in the user group and club list that used to appear on page 703 whilst Martin was away on a course and Keith was standing in on his behalf.

Many months went by and NEXT was expanding more and more until things came to a head and Martin Cooper decided that NEXT was taking too much time, the magazine had to be cut down to what you can see now on the screen. It is a shame that all good things came to an end.

As an example, here is an extract from one of Keith Hazelton's articles about computer fanzines and how the mailbox was set-up.

```
----------------------------------------
PRESTEL             77a             0p:
FROM    019998731                     :
TO MBX 219999677              21.52   :
NEXT                                  :
                                      :
    PAPER ROUND - Page 1, Line 1 to 9. : < This is the header to
1   This month a selection of hints and: indicate the number  of
2   tips for small magazine editors.   : pages sent.
3                                     :
4   Thanks to all those who have       :
5   written in, and in particular to   : Notice  also  that  the
6   Phil Bridges, secretary of AMRAC   : number   of   characters
7   User, Andy Dunn of SoCo NEWS and   : per line is 35 and that
8   Ewan Dalton of The Federation of   : the separation line  is
9   Small Computer Magazines.          : also numbered to  break
----------------------------------------  the text up to  a  more
PRESTEL             77a             0p: readable paragraph.
FROM    019998731                     :
TO MBX 219999677              21.53   :
NEXT                                  :
                                      :
    PAPER ROUND - Page 1, Line 10 to 18:
10                                    : < Line 10 is the centre
11  It seems that most publications    : of the teletext screen.
12  start off as photocopied affairs   : Additional   space   is
13  and only progress to conventional  : allowed    for    other
14  printing when finances permit.     : information    to    be
15                                    : included. e.g.  contact
16  Remember, at first your readers    : address   or  price  of
17  will primarily be interested in the: software reviewed.
18  content, NOT the quality.          :
----------------------------------------
```

Lets hope that one day the BBC will allocate more resourses to CEEFAX, then NEXT can be restored to its former glory. In the meantime Channel 4, Oracle, page 568 now has micro charts + news and reviews every Thursday and Friday.

KITCHEN SCHEDULER

By: Emma Louise.

This program was written one day to help with the Christmas dinner but I am sure it can be modified for other uses. All that it does is take a list of the separate items that go to make up a meal, then it asks how long each will take to prepare and cook. After some calculations it then prints out a schedule or time table so that everything can go smoothly. It avoids having to prepare two items at once and it ensures that the all the parts of the meal are ready at the same time.

The entire program is written in BASIC and should be fairly simple to understand, but if not a few hints on what is happening follow.

The program starts at line 9000 and the items are held in the array i$() the preparation and cooking times are held in the arrays p() and c().

The times are entered as a decimal number with the hours before the decimal point and the minuts after the point. (Only 59 minuts are allowed) so 13.45 should be entered for 1.45pm. All the times are converted and held as the number of minutes since midnight.

Text justification is performed, when the times are printed out, to give neat columns. All the calculations are done on the two dimensional array d() first they are sorted into cooking time order then the preparation times are adjusted if needed to prevent any overlapping times.

Finally your time table is printed out either to the screen or the printer.

This program has examples of sorting arrays, printing numbers aligned around a decimal point, splitting long lines of text at word endings and real number comparisons. It should work with all versions of the Spectrum and does not rely on any peripherals, although a printer is very useful.

Last but not least it should ensure that you never have burnt spuds with raw sprouts ever again.

```
 10 GOTO 9000
 20 REM split m$ nicely
 30 IF LEN m$<33 THEN RETURN
 40 IF m$(32)=" " THEN LET m$=m$( TO 32)+m$(33 TO ): RETURN
 50 LET z$=m$( TO 31)
 60 FOR l=LEN z$ TO 1 STEP -1: IF z$(l)<>" " THEN NEXT l
 70 LET z$(l TO )=" "
 80 LET m$=z$+m$(l TO )
 90 RETURN
```

```
100 REM print a time
110 GOSUB 400
120 LET t$=STR$ t
130 LET po=0
140 FOR p=1 TO LEN t$: IF t$(p)="." THEN LET po=p
150 NEXT p
160 IF NOT po THEN LET t$=t$+"."
170 IF LEN t$<3 THEN LET t$=" "+t$: GOTO 170
180 IF t$(3)<>"." THEN LET t$=" "+t$: GOTO 180
190 IF t$(2)=" " THEN LET t$(2)="0"
200 IF LEN t$<5 THEN LET t$ = t$+"0": GOTO 200
210 PRINT #chan;t$(1 TO 5);" ";
220 RETURN
300 REM convert t->r
310 LET ti=INT t
320 LET td=t-ti
330 LET r=100*td+60*ti
340 RETURN
400 REM convert r->t
410 IF r<0 THEN LET r= r + 1440
420 LET ti= INT (r/60)
430 LET td=(r-60*ti)/100
440 LET t=ti+td
450 RETURN
500 REM accept a time
510 GOSUB 20
520 INPUT (m$+" ");t
530 IF t<0 THEN GOTO 500
540 IF INT t > 24 THEN LET m$="Invalid hours": GOTO 500
550 IF (t-INT t)>=.6 THEN LET m$="invalid minuits": GOTO 500
560 GOSUB 300: RETURN
600 REM get a list of items
610 DIM i$(50,32)
620 FOR i=1 TO 50
630 LET m$="Enter item # "+STR$ i
640 INPUT (m$+" "); LINE a$
650 IF LEN a$ THEN GOTO 700
660 PRINT #0;AT 0,0;"Any more items?",
670 IF INKEY$="N" OR INKEY$="n" THEN GOTO 720
680 IF INKEY$="Y" OR INKEY$="y" THEN GOTO 640
690 GOTO 670
700 LET i$(i)=a$
710 NEXT i
720 LET i=i-1: RETURN
730 FOR l=LEN a$ TO 1 STEP -1: IF a$(l)=" " THEN NEXT l
740 LET a$=a$( TO l): RETURN
800 REM get prep & cook times
810 DIM p(50): DIM c(50)
820 FOR n=1 TO i
830 LET a$=i$(n): GOSUB 730
840 LET m$="Enter preparation time for "+a$
850 GOSUB 500: GOSUB 300: LET p(n)=r
860 LET m$="How long will "+a$+" take to cook"
870 GOSUB 500: GOSUB 300: LET c(n)=r
880 NEXT n
890 RETURN
900 REM get meal time
910 LET m$="What time do you want the meal?": GOSUB 500: LET time=r
```

```
 920 RETURN
1000 REM calc cook times
1010 DIM d(50,4)
1020 FOR n=1 TO i
1030 LET d(n,4)=n
1040 LET d(n,3)=time-c(n)
1050 LET d(n,2)=d(n,3)
1060 LET d(n,1)=d(n,3)-p(n)
1070 NEXT n
1080 REM sort em into cook time
1090 LET done=1
1100 FOR n=1 TO i-1
1110 IF NOT (d(n,3)<d(n+1,3)) THEN GOTO 1160
1120 FOR m=1 TO 4
1130 LET tmp=d(n,m): LET d(n,m)=d(n+1,m): LET d(n+1,m)=tmp
1140 NEXT m
1150 LET done=0
1160 NEXT n
1170 IF NOT done THEN GOTO 1080
1180 RETURN
2000 REM adj prep times
2010 LET prep=d(1,2)
2020 FOR n=1 TO i
2030 IF d(n,2)>prep THEN LET adj=d(n,2)-prep: LET d(n,2)=d(n,2)-adj:
 LET d(n,1)=d(n,1)-adj
2040 LET prep=d(n,1)
2050 NEXT n
2060 RETURN
3000 REM printout
3010 LET now=time
3020 LET e=0: LET w=0
3030 FOR n=1 TO i: FOR m=1 TO 2
3040 IF d(n,1)<now THEN LET w=n: LET e=1: LET now=d(n,1)
3050 IF d(n,3)<now THEN LET w=n: LET e=3: LET now=d(n,3)
3060 NEXT m: NEXT n
3070 IF NOT w THEN GOTO 3500
3080 LET r=d(w,e): GOSUB 100: PRINT #chan;TAB 6;
3090 IF e=1 THEN LET r=d(w,2): PRINT #chan;"to ";: GOSUB 100
3100 PRINT #chan;TAB 16;"prepare " AND e=1;"cook " AND e=3;: LET a$=
i$(d(w,4)): GOSUB 730: PRINT #chan;a$
3110 LET d(w,e)=time
3120 GOTO 3000
3500 LET r=time: GOSUB 100: PRINT #chan;"Dish it out and eat"
3510 RETURN
9000 CLS : GOSUB 600: GOSUB 800: IF NOT i THEN CLS : PRINT "Are you
not hungry then": PAUSE 200: PRINT "press any key to continue": PAUS
E 0: GOTO 10
9010 GOSUB 900
9020 CLS : PRINT FLASH 1;"Please wait": GOSUB 1000: CLS
9030 INPUT "Screen or Printer?";t$
9040 LET chan=2: IF t$="p" OR t$="P" THEN LET chan=3
9050 PRINT #chan;"THE SPECTRUM COOKING SCHEDULER"''
9060 GOSUB 2000
9070 GOSUB 3000
9080 PRINT #0; FLASH 1;"Run again?"
9090 IF INKEY$="y" OR INKEY$="Y" THEN RUN
9100 IF INKEY$="n" OR INKEY$="N" THEN STOP
9110 GOTO 9090
```

YOUR LETTERS

Dear Editor, *STAR*LETTER* *STAR*LETTER*

I would like to use your letters page to ask a question. What has happened to all the simple type in programs like those that use to appeared in magazines two or three years ago. I enjoyed typing them in, I learnt a lot about Basic and programming techniques from them. Its also true to say that the simple games often contained more 'Play Value' than many of the machine code arcade games. Cant FORMAT resurrect this dead art-form? I cant be the only one who likes to type in programs.

Yours Sincerely, Dave Sowton.

We can only print what we get sent in Dave. Most of what we receive are utility programs but perhaps your appeal will bring forth some small games. We can but wait and see. ***Ed.***

Dear Editor,

Just a quick comment on the +2a vanishing trick. After reading 'News on 4' in your December issue, I noticed that several shops here in Germany are selling the Spectrum, even though they have never sold Sinclair machines in the past. In fact they are selling the +2a in its shining black case. I wonder, is this a new form of DUMPING?

Yours Sincerely, Ian Spencer.

Dear Editor,

May I express my thanks to all your writers for making FORMAT such an enjoyable read each month. I may not understand everything thats printed (machine code is beyond me at the moment) but the light is beginning to dawn on a lot of things.

Yours Sincerely, Brian Candy.

Dear Editor,

With reference to the letter from Dave Morgan (Vol 2 #5) regarding the poor picture on his Plus 2. I can assure him that he is not alone in this problem. I bought a Plus 3 from Dixons which would not tune in correctly on any of the four TVs on which it was tried. An exchange was obtained with no improvement. Another exchange Plus 3 was better but still far from 100%. The trouble in my case seems to be the 6MHZ sound to vision carrier separation from the TV modulator. Should this not be correct it is impossible to tune in correctly on a TV receiver. Some TVs seem to be better than others in tolerating these errors.

Yours Sincerely, A.Fletcher.

There really is no excuse for this problem, its just poor quality control on the part of Amstrad. Most repair companies will retune the TV output but why should you have to pay them to cure Amstrads poor workmanship? If more people returned their computer as you did Sugar would be forced to take steps to sort out his production standards. ***Ed.***

Letters printed may be edited for length or clarity. The writer of each months STAR LETTER wins an EXTRA 6 months subscription.

MIDI

MIDI FOR BEGINNERS PART 6 - BY RAY ELDER.

Well here we are at the end of this series, I may well write a few follow up articles, but to end with I offer some ideas and techniques which you may find helpful.

GENERAL HINTS.

When I use an instrument such as the Yamaha FB01 or Casio CZ101 I tend to try combinations of sounds. With the Casio in Mono mode (four independant 'instruments' of one voice) or when using the XRI Step time sequencer you are limited to single notes which cease to sound before or at the moment another note is played. This can cause the song to be very 'clean' and somewhat robotic when replayed. This is fine for any instrument which has this kind of staccato feel.

However many instruments such as piano or guitar normally has some resonance continuing after the note has been released, the only way to compensate for this is to use an instrument which has more than one note per sound, ie. the CZ101 in 4 or 8 note mode.

If you have such an instrument as the FB01 or D110/MT32 then you can assign a selected number of notes to each section. For instance, I often use two notes for the bass voice on the FB01, this allows for a slight overlap of notes when playing a supposedly single note bass run and can compensate for the overresonance. This becomes my 'master' bass sound.

The sound can be further enhanced by adding a single note, secondary, different bass sound to support it. I may use an upright bass sound as master and add a slap bass note as support.

If the bass is particularly important (as it usually is for me) I often use yet another note in another instrument configuration and detune it by +/-8 for added depth to the sound.

By now I have used up half the avialable notes just for one instrument! To get over the rather 'clipped' staccato effect of the music I try to layer music so that bars overlap, this is easy to do either in real time or in step time. Take for instance, the classic 12 bar rock /blues pattern. A suggested overlay might be like that in Fig 1.

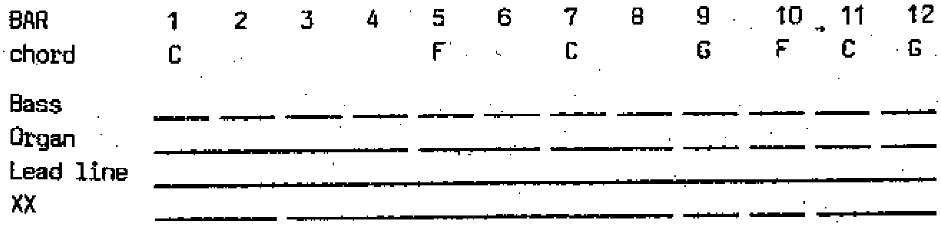


Fig 1.

As the lead line may well be staccato in its nature you can bind the bars together by subtle (important word) use of harmony notes as shown by the line XX. A possible, but not very sophisticated, harmony could be a G for 1 & 2, C for 3-8, D, C, G for 11 & 12.

This would help to bind the bars together although volume and octave and voice would all have to be carefully chosen so that it was not intrusive.

BASS.

I mentioned some bass techniques earlier, but in general I find this to be an important element of many songs. Careful choice, and feasible thanks to MIDI sequencing, much trial and error replaying may be needed to find the best sound. I often find that, on the FB01 at least, some instruments which are not specifically designated as bass give effective bass sounds, piano, guitar, brass, cello are all worth investigating in your search for the best bass sound for a particular song.

Although the bass may well be used for effect by playing across the rhythm, in general the bass plays in time with the bass drum. A tight bass/drum combination can really move a song along.

SOUNDS.

No matter how brilliant the piece of music or how well programmed/played, if the instrument sound itself is wrong then you will never be satisfied with it. If you are an experienced sound synthesist then you already know what to do to get the sound you require, if you are like me then you may find a sound that is close to what you want and then 'tweak' it by altering some of the parameters to make it suitable.

It is here that Editor programs become useful, however you can, provided you are using a synth and not a preset keyboard, adjust the parameters from the instrument's panel.

Here are a few simple hints, all the terms I use are not particularly technical, more descriptive of the general sound:

To increase/decrease sound 'movement'
Use instrument's or external chorus effect.
Detune oscillators or add slow vibrato.
Add slow LFO modulation to the filter cut off frequency.

To make sound 'brighter'
Select waveform with harmonics (Sawtooth etc.)
Increase VCF cut off frequency, filter envelope sustain or amount.
Increase high frequencies with external EQ.

To 'fatten' a sound
Use a second note/oscillator one octave below.
Add slow chorusing.
Increase low to mid EQ.
Add some resonance and tune the filter to around 500Hz or lower.

To make a sound 'hollower'
Change waveform to triangle or square.

To 'dull' a sound
Change waveform to a triange or sine.
Decrease filter cut off frequency, filter envelope sustain or

amount.
Decrease resonance.

To add percussion
Set VCA or VCF envelope generator attack time to zero and reduce the initial decay time. You may also have to adjust the sustain. Try adjusting the VCF first, then modify the VCA if necessary.

To increase 'sharpness'
Increase filter resonance.

To 'thicken' a sound
Change waveform to Sawtooth or similar.
Use internal or external chorus.
Add short echo.
Add slight amount of distortion with an effects unit.

To enhance 'sheen'
turn up the filter resonance to around 60%
Add an harmonic booster effect unit.

To get a 'thinner' sound
Change waveform to triangle or sine.
Switch of one of the oscillators if dual oscillators are used.
Turn off internal chorus.

To get a 'warmer' sound
Decrease filter cut off.
Decrease resonance.
Boost bass/mid EQ.
Reduce vibrato speed.
Change waveform to sawtooth.

To get a 'wetter' sound
Add chorus.
Decrease filter cut off
Change waveform to sawtooth or sine.
Add reverb or delay.

Much of the problem with sounds is not so much the sound texture but the inappropriate way in which it responds to rhythm and this is very much dependant on the envelope generator settings, in general:

Long release/attack times are best used for legato type pieces.
Short attack and/or release times are best for up tempo music.
Increase release times for slower paced songs to make for a fuller sound and add 'body'.

As you can see I have referred to 'external' effects units in this outline, yet more expense! I find that you can purchase a wide variety of these ranging from the small foot operated guitar type box from around twenty pounds to studio racks at five hundred pounds plus - with many units of varying quality and price in between.

My personal order of priorities is as follows:

EQ. (Equilisation) -
A modern term for what we oldies used to call tone controls. The

old Bass, Middle and Treble knobs have now been replaced by an array of knobs often controlling specific frequency ranges marked in Hz.

These now range from 3/4 knob jobs to multi-slider machines commonly known as Graphic EQ's. An absolute essential to be included somewhere in your sound system.

REVERBERATION UNIT -

. Spring ones cost about fifty pounds to two hundred pounds, good for orchestral and non percussive sounds. not much good for drums where they sound 'boiingy'.

Digital units cost from two hundred pounds up. clean, very effective, very versatile.

ECHO UNIT -

Mechanical or tape based, good and are priced around a hundred pounds up. Lacking in some areas or control and can be noisy with tape hiss.

Digital units cost from around a hundred and fifty pounds up. Excellent control and most are very clean.

CHORUS -

For most of my work I use a guitar type here, cheap at around twenty five pounds but lacks sophistication of the more expensive studio units.

Flanger/Distortion/etc.

There are many such exotic effects units on the market, some musicians use them a lot, but personally I have not found them much use for the music I tend to produce.

Of course all these units can be purchased for much less in second hand state and the ones I have, have worked perfectly for several years now.

And so we come to the end of this series. I have enjoyed writing this, a chance to collect all my ideas and to get them in order. Also a chance to hear from other musically minded computer users. I hope you have enjoyed these articles and those of you who have purchased my tape of audio recordings (Last Plug!) still available for only £3.95 at the address below, may have been inspired to produce bigger and better things themselves.

I would love to hear from you, ideas, tips, queries, demo tapes of your own music, or indeed anything on this topic. Please write to me direct (enclosing cheque or PO if you want a tape) at 1 Periton Court, Parkhouse Rd. Minehead, Somerset. TA24 8AE.

PS. I have had an enquiry from a reader who wanted to know of any program that will print 32 bars of music on an A4 piece of paper using his Spectrum. I haven't any knowledge of such a beast, the nearest being a £125 Atari ST program (and £300 for the computer). He has Music Typewriter and is thinking of buying XRI's Step/Real time sequencers. I don't suppose any reader has modified these or any others to print out in the manuscript form he requires? If so, we would both be very grateful if you could drop me a line I'll pass on your name(s) and address(es) to him. Thank you.

See you soon.

ADVENTURE CORNER

By: **Paul Rigby.**

Last month I completed a brief round up of all the popular, and not so popular, utilities and languages that you, the adventure author, could use to create an adventure. As I mentioned, it appears that Utilities are the most popular method of producing an adventure, especially for those non-programmers among us. Two of the Utilities mentioned below have even been converted to the 16-bit market. The GAC has been re-worked to be sold as the STAC for the ST and the PAW should be available for the PC as you read this. But what, exactly is an Adventure Utility?

Well, very simply, a Utility is a collection of interrelated databases, or rather, a database of interlinked files. Each file, within the database, contains similar data such as messages, screen colours or graphics which have been previously created by the author. So, for example, if, in your adventure, you wished to step into a kitchen the program would check the location number given for the kitchen. A pointer would then leap from the location number to the atmospheric, descriptic text you wrote earlier, situated in the "Description File" and show it, in all its splendour, on the screen. If a graphic had been created for this location another pointer, again from the original location number, would retrieve the relevent graphic from the "Graphic File" and show it with the description. Another file is then checked, the "Object" file. The pointer quickly scans this file picking all of those objects which should be present in our Kitchen. So, when you actually play the adventure, all of this information is displayed on screen at once.

If an object is examined another pointer, this time from the "Object File", retrieves the relevent description of the object from the "Object Text File" and displays it on screen. When any words are input from the player the "Vocabulary File" is checked to find a match for the associated verbs and nouns. Of course if any input is not understood a relevent system message is extracted from another file and shown on-screen such as, "I don't understand."

The above is a simplified description but the general principles are the same for all Utilities. The differences between each Utility are normally the amount of extra features presented within the overall package and, indeed, how the package has been put together in the first place. What follows is a personal evaluation of the three most popular Utilities on the market today and a recommendation for any reader who is contemplating buying one.

The oldest of the three and the first by Gilsoft is that old faithful The Quill. Many adventure authors were weaned on adventure programming with this excellent Utility. Over the years it has grown with a number of programs which have been tacked on to increase its capabilities. The most popular being the Press, Patch and

Ilustrator. Which generally increased The Quill's flexibility by allowing graphics to be used within the game, the inclusion of sound effects, improvements in the screen presentation and so on. The main advantage of the Quill is its simplicity. This is because it uses normal English as its command structure. However, it has become rather outdated compared to the other Utilities and other adventures on other computers. The use of flags, for example, is rather restricted (a Flag signifies status such as whether your Lamp is on or off) and the parser is limited. However, if you work hard with The Quill good adventures can be produced, some authors have manipulated The Quill itself to incorporate different features. In fact some excellent adventures are still being marketed using it.

The second Utility in this evaluation is the Graphic Adventure Creator, popularly known as the GAC, from Incentive. This second generation Utility is fairly hard to come to terms with initially. It contains some very powerful features, though. One example being the versatile method of flag manipulation, which is very important within an adventure. However, there are disadvantages that let the package down. A poor compression routine was initially included. When it is used some words that are deleted from the database still remain in the memory. Also, the memory that the GAC allows for free use is relatively small for the author to play with. The addition of the GACPAC by Essential Myth, a sort of Utility for an Utility, created some order within the chaos by increasing the flexibility of the GAC. A new compression routine brought a great improvement over the original example, also methods were incorporated to increase response time and present a friendlier user-interface with the GAC.

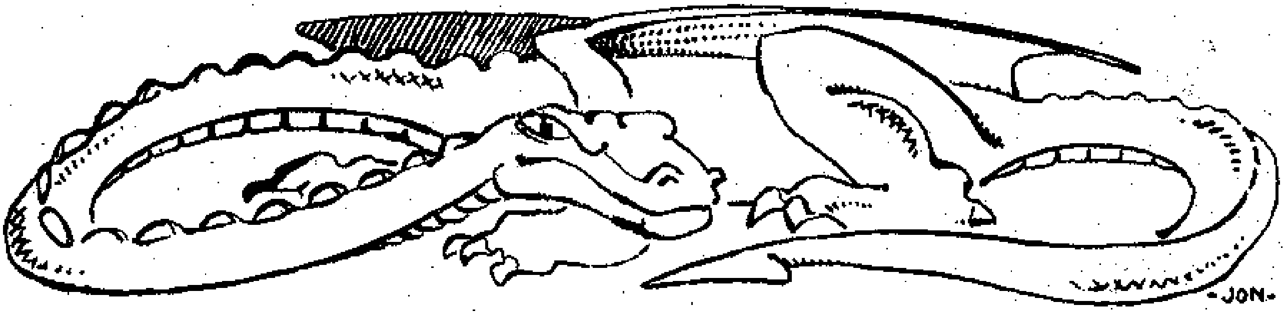
The third Utility in this overview is the latest product from Gilsoft. The Professional Adventure Writer, commonly known as the PAW. It has all of the simplicity of the Quill with much of the power present in the GAC - plus a whole lot more! Probably the best feature of the PAW, and any other Utility on the market, is the EXTERN command. This allows the user to bring in external code, such as Basic and Machine Code, to be used within the adventure. This one feature should sweep away all of those criticisms of Utilities which say that they all look the same. The EXTERN command allows for personal features to be written and brought in. Even if you lack the knowledge to use such a feature to its full the PAW has enough features of its own for you to be getting on with which should still maintain an element of variety, such as the ability to swap character sets mid-screen. However, the one disadvantage with the PAW is that it is really only suited to the bigger 128K Spectrums. A great deal of fiddling about is needed with the 48K Spectrum. Constantly having to use Overlays, for example, can be frustrating.

So which is it to be? Well, considering the <u>initial</u> version of the PAW had all of the features of the <u>mature</u> version of the Quill, along with it's suite of additional programs, I cannot really recommend the Quill for the new user. Why bother taking time to learn it when more advanced programs are readily available? On the other hand the GAC seems to be a failed attempt to create the ultimate Adventure Utility. In some ways it is more powerful than the PAW but these features are poorly implemented. The PAW has a wider range of useful features which have been put together in a more sensible manner so that it works better as a complete package and not, as in the GAC, as a mixed set of powerful and unfinished programs. Also, and of particular significance to Format readers, the PAW . is available for the Plus D/Disciple in the form of a tape which is transferred to disc (so it does not matter whether you use 3.5 inch or five and a quarter inch discs), neither the Quill or the GAC can be used with the Plus D/Disciple. So this version of the PAW will result in a greater amount of immediate memory being available and also, of course, it is much faster! Finally, Essential Myth, the creators of GACPAC, are adventure authors themselves. They stated that they could only use the GAC to complete their first excellent adventure, " The Book of the Dead.", with the additional use of their GACPAC. Further, after they had spent so much time improving the facilities of the GAC they announced that they would be using the PAW for their next adventure due to its overall superiority! The result is another excellent adventure, this time using the PAW, called "Dr.Jekyll and Mr.Hyde".

Consequently, I recommend the PAW as the Utility to go for. It is relatively easy to learn, powerful, can be disc-based and is, well, more Professional! However, if you are a Quill, GAC or PAW user who is now jumping up and down in rage at my conclusions, if you would like to tell the world of any particular features of your Utility that you love or hate, or if you have added your own features (big or small) to your Utility, whatever it may be, then please write to me, care of Format.

To end this month I would like to continue the subject of mazes which I began last month. If you remember, I described one method of completing a maze, that of dropping objects from your inventory to establish where you are within the maze. However, this method has its drawbacks in certain adventures. The adventure may be designed so that you do not have enough objects in you inventory to cover all of the locations. One way to combat the latter problem is to carefully examine the location description or associated graphic. Sometimes they may differ very slightly between each location. So, although each location description is the same, there may be a comma or full stop in a different position or an extra word inserted and so on. Similarly, in the graphic, a minor feature, such as a flower, may be added, taken away, growing in a different direction or shown in a different colour. Thus, you can include these irregularities on your map. I will include other tips in future issues. If you have any tips of your own why not take pity on some desperate adventurers and write in. I will include the most useful ones in a future Adventure Corner.

Next month Adventure Corner whisks you away to times gone by, to Caves and, well, more Caves! How was the Original Cave adventure created? Who were Crowther and Woods? Adventure Corner will reveal all! So, light your brass lantern and keep your eyes peeled for Adventure Corner next month!



SMALL ADS. SMALL ADS.

YOUR ADVERT

Buying, Selling, Pen Friends, Clubs, etc.

This space is reserved for you. Any PRIVATE advert, up to 30 words (subject to acceptance), will be printed FREE in the next available issue. Any software sold must be original copies, with full instructions and in working order. The publishers will not be held, in any way, responsible for adverts in this column. Trade advertisers should contact the publisher for a rate card.

* - * - * - *

BACK ISSUES

For members who have missed past issues of FORMAT (or perhaps worn theirs out through constent use) we run a back-issue service.

The cost is £1 per issue (£1.25 overseas) incl p&p. Your copies will be sent out with your next monthly issue of FORMAT (provided we receive your order at least a week before). Make cheques (drawn on UK bank or Euro-Cheques, P.O., cash) payable to FORMAT.

AVAILABLE ISSUES

Vol 1 Issues #7 (Feb 1988) - to -' #12 (Jul 1988).
Vol 2 Issues #1 (Aug 1988) - to - #6 (Jan 1989).

Note:- Issues 7 to 11 of Vol 1 are now in short supply.

Please WRITE YOUR ORDER ON A SEPARATE PIECE OF PAPER. DO NOT include letters with your order as this will cause delays.

Remember to quote your membership number or we wont be able to send you your order.

SPECTRUM
ROM ERRORS

or WHERE THE PROGRAMMER DOZED OFF.

By: Francis G Miles.

I have been working recently on the "Spectrum ROM Disassembly" by Drs Logan and O'Hara, and have been amusing myself by devising demonstrations of the various "Spectrum Crashes" which are alluded to - sometimes very cryptically - in their notes. I hope readers will be interested in my findings.

1. The simplest and most obvious "crash" - and it can be quite serious - can be produced by typing **CLS: PRINT CHR$ 8+"hello"** (CHR$ 8 is the "backspace"). This prints "ello" at the top of the screen, but the "h" has got backspaced right off the screen and (somehow) into the attributes and system variables! I am not quite sure which ones (and the weird arithmetic of the display subroutine **PR ALL** at location 2943 (OB7F hex) makes it quite hard to figure out) but you are likely to find that all your variables have disappeared ("Variable not found") and cannot be replaced, which suggests that **VARS** (23627) is one of them. Backspacing other character codes into the system variables will cause similar forms of chaos.

This error is in **PO BACK 1** which starts at 2595 (OA23h), the programmer intended to inhibit a backspace from position 0,0 on the screen - 24,32 (18,20hex) in the "position value" system of the ROM - so he programmed a jump if the line number is 18h. In the depth of his slumber he didn't notice that this should have been 19h, so this disastrous backspace remains possible. (If you type **PRINT AT 0,1; CHR$ 8+"hello"** the backspace is ignored.)

2. One which probably annoys quite a lot of BASIC users is the **REM** colon error: if you try to put a heading on your subroutines like **1000 REM subroutine: Display Table** you will find your input comes out as **1000 REM subroutine: DIM isplay Table** In REMs, the next keystroke after a colon is always interpreted as a keyword, never as a letter. The only ways I know to sidestep this problem are:

2.1 Type the letter first, then backspace and put in the colon.

2.2 Put some other character after the colon and delete it after you've typed in the letter - e.g. type the letter twice. These work all right, but they're quite tiresome.

2.3 Put all your REMs in quotes, or just with a quote " at the start. Then you don't have to think about it. This error is in **OUT CHAR** at 6455 (1937h): during key input, a colon is always read as an end-of-statement marker and switches the mode to K mode, even in a REM, unless it is in quotes. Unlike the other errors described here, it is difficult to see just what could be done about this one: probably a REM flag would be needed.

3. Another interesting, but much less troublesome error is

demonstrated by typing two lines with consecutive line numbers as the last two lines of a BASIC program, e.g.

```
9998 STOP
9999 STOP
```

Now type 9999 and ENTER, which deletes the last line; and press EDIT. Line 9998 appears for editing with its current line cursor ">" - and you can't get it back into the BASIC till you delete this cursor. (Actually, pressing "up arrow" or "down arrow" and then EDIT again will get rid of it).

This error is in **ED EDIT** at 4009 (OFA9h). The line is printed out to the lower screen by a call to **OUT LINE** at 6229 (1855h), the same subroutine which prints out program lines in the upper screen for listings; and this is set to include the current line cursor if the number of the line being printed matches **E PPC**, the current line number. So the programmer thought he could very neatly avoid the printing of the cursor in the editing line (which is always the current line or the next after it, he thought sleepily) by temporarily reducing E PPC by one. But if you delete the last line of a program, **E PPC** is left for the time being with its number, and if you press EDIT it brings down the last line left in the program; and if it has a line number one below **E PPC, OUT LINE** will obediently print the cursor in it.

It would have been better to set the high bit of **E PPC**, making it an impossible line number, 32768 or more, and the unset it again afterwards.

4. If you type **SAVE "string" DATA s$()** s$ is supposed to be a string array, which has been declared by a DIM statement. If you unfortunately do this with an ordinary string s$, which has not been dimensioned, it will save on tape all right, and load back from tape all right; but if you then type PRINT s$ you get "Subscript wrong", and nothing you can do (from BASIC anyway) will allow you to get at s$. (You can't do this with DISCiPLE and PLUS D - Bruce Gordon obviously knows about this one. It won't accept the original SAVE d1 DATA command unless s$ is dimensioned).

This error is in 0652 **SA DATA** at 1618 (0652h), which (I think) the programmer wrote SET 7,C when he meant to write BIT 7,C. Asleep at the switch again.

5. Quite interesting, but not very troublesome, is the error shown by **100 PRINT "Result = "+STR$ n** If you give n any value (absolute) of one or more, this works perfectly correctly: eg if you type **LET n= PI: GOTO 100,** it prints out Result = -3.14159 But if you give n a value (absolute) less than one, it won't print the "Result =" message: **LET n=1/7: GOTO 100** prints out 0.1428571.

This error is in **PF SMALL** at 11812 (2E24h); Dopey has left something on the stack that he shouldn't have. When the expression **"Result = "+STR$ n** is evaluated by the SCANNING loop (9647 (25AFh)), first the string parameters of "Result =" go on to the calculator stack then the op code and priority for "+" go on the machine stack then the string parameters of the value of **STR$ n** go on the calculator stack on top of the other string parameters (but unfortunately Dopey has left a zero between them, if n comes out small) and now "+" is evaluated by **STRS-ADD** (13724 (359Ch); off the calculator stack come the first set of string parameters, and they're concatenated with the next lot down; which ought to be the string parameters for "Result = ", but instead it gets the null string indicated by the spare zero on the stack.

Moral: always clear the calculator stack.

Another moral: there is always one more bug in a program, no matter how long you spend debugging it.

* * * * * * * *

PARTS LIST FOR LAST MONTHS RS232 INTERFACE

Here is the list of components we managed to leave out of last months article.

Resisters

R1	= 10k
R2,3,4	= 470
R5 to 11	= 10k
R12,13	= 4k7
R15	= 470

Capasitors

C1	= 10nf
C2	= 100nf
C3	= 10nf
C4,5	= 10uf

ICs

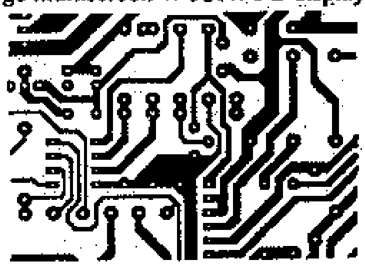
IC1	= 74LS00
IC2	= 74LS04
IC3	= 4040
IC4	= 6402
IC5	= TL071
IC6	= ICL7660

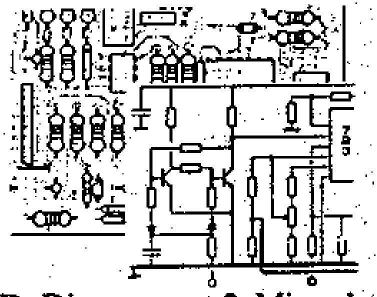
Other Parts

D1	= IN914
TR1	= VN10
X1	= 2.4576X

Sorry again for not printing it last month. Hope you enjoy building this project now you know what you need.

More hardware soon.





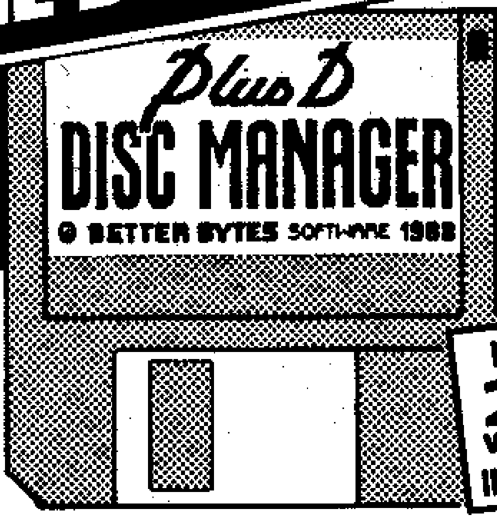
· NAMES & NUMBERS ALL YOUR DISCS ·

EASY TO USE

THE ORIGINAL

DISC MANAGER

FOR THE DISCIPLE & PLUS D



NEW UPGRADE VERSION

A MUST FOR ALL DISCIPLE & PLUS D USERS

INCLUDES THE BEST EVER 2 WAY SEARCH IN M/CODE FOR SPECTRUMS

GET TOTAL CONTROL OF YOUR DISC COLLECTION

CUSTOMERS THROUGHOUT THE WORLD – HERE'S WHAT SOME OF THEM SAY...

'IMPRESSIVE' A.S., NEW ZEALAND

'MARVELOUS GRAPHICS' I.A.R., RENFREWSHIRE

'A DELIGHT TO USE' F.W.J., CHIPPENHAM

'VERY USEFUL' R.C.M., NETHERLANDS

'EXCELLENT' R.H. MIDDLESEX

'ESSENTIAL' R.W. CALIFORNIA

PROBABLY THE LARGEST PROGRAM EVER WRITTEN FOR THE SPECTRUM! COMPRISES

12 PROGRAMS

6 MAIN – 6 SUPPLEMENTARY
A total of 36 files supplied on disc – over 250K of Program Magic!

The NEW upgrade version of the original DISC MANAGER is the most powerful programme ever written for the DISCIPLE/PLUS D.

Designed to take advantage of Disc Drive ownership, the Manager keeps track of all the programs on all your discs, offering unrivalled benefits and features.

- Storage of up to 27,000 records on one Disc, or 79,920 total. Random File Access.
- Name & Number Discs with fast Autonumber and user pre-defined titles features.
- No typing in of Data. Press a key and Discs are automatically added to appropriate catalogue.
- Fastest ever M/Code Search. 2 modes – Search and Load or Search and List all occurances, then select program to load.

STOP PRESS NOW AVAILABLE **NEW** DISC ORGANISER

Recover Erased Programs · Re-organise Directory · Repair Sectors · Wild Card Copy, Erase or Re-name · Supplied on Cassette ONLY **£5**

- Special program to print the contents of all your discs with fast sort option. Can selectively print by disc type or number.
- List contents of any disc with ease.
- Multiple erase or rename options.
- Plus many other unique features.
- Comes with 16 page manual and demonstration catalogues.
- Operates with 48K or 128K Spectrums.

Send for the DISC MANAGER today... and you'll soon wonder how you ever Managed without it!

NORMAL PRICE £14.95

SPECIAL INDUG PRICE ONLY

£12.95 INC. P&P

OVERSEAS ORDERS PLEASE ADD £1.50

UPGRADES Existing users can upgrade to the NEW Version.
Send old Manager Disc plus £5.00

BETTERBYTES

10 SPITTAL TERRACE · GOSFORTH
NEWCASTLE UPON TYNE NE3 1UT · TEL: (091) 285 6185

· SPECIAL PRINT-OUT PROGRAM · 16 PAGE MANUAL · FLEXIBLE · GREAT GRAPHICS

PERSONALISE YOUR DISCS · AUTOMATIC CATALOGUER · FAST SEARCH & LOAD · PULL DOWN MENUS ·

· SUPPLIED ON 3·5 OR 5·25 DISC · WRITTEN SPECIALLY FOR DISCIPLE/PLUS D ·

PLEASE STATE YOUR DRIVE TYPE, SIZE, ETC, INDUG MEMBERSHIP No. AND IF REQUIRED FOR DISCIPLE OR PLUS D